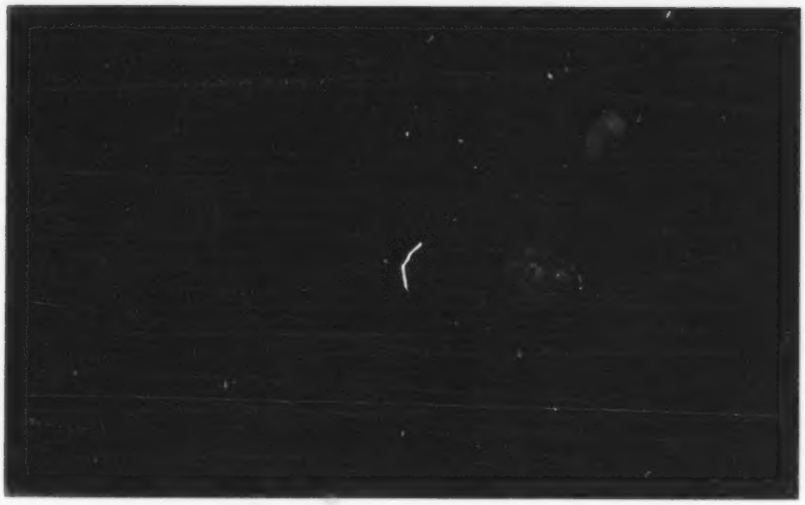
CENTRE DE RECHERCHE
SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE



CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A,
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: (21) 05-561325
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

CRT-732

A TWO-COMMODITY FLOW FORMULATION FOR THE TRAVELING SALESMAN AND THE MAKESPAN PROBLEMS WITH TIME WINDOWS

by

André Langevin[1,2,4]
Martin Desrochers[1,2,4]
Jacques Desrosiers[1,3,4]
François Soumis[1,2,4]

1 Groupe d'études et de recherche en analyse des décisions (GERAD), 5255, avenue Decelles, Montréal, Québec, H3T 1V6.

2 École Polytechnique de Montréal, C.P. 6079, succursale A, Montréal, Québec, H3C 3A7.

3 École des Hautes Études Commerciales, 5255, avenue Decelles, Montréal, Québec, H3T 1V6.

4 Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale A, Montréal (Québec) H3C 3J7.

Centre de recherche sur les transports - Publication #732
Cahier du GERAD - G-90-44

Novembre 1990

91-00960/

Dépôt légal - 4e trimestre 1990

Abstract

We present a new two-commodity flow formulation for the traveling salesman problem. Each commodity corresponds to a resource that is either distributed or picked-up along the tour of all nodes. This formulation is particularly well-suited to handle time window constraints; the resource used is then the time. This formulation can be extended to the makespan problem. For a n-node problem, the linear relaxation of the formulation involves only $O(n)$ constraints and $O(n^2)$ variables. Implementation issues are discussed and numerical experimentations have been realized for problems of up to 40 nodes.

Key words: Routing, Scheduling, Traveling Salesman Problem, Time Windows.

Résumé

Cet article propose une nouvelle formulation avec deux flots complémentaires pour le problème du voyageur de commerce. Les deux flots correspondent respectivement à la distribution et à l'accumulation d'une ressource le long des arcs décrivant le tour. Cette formulation est particulièrement bien adaptée pour le problème avec fenêtres-horaire: le temps constitue alors la ressource. Cette formulation s'adapte facilement pour optimiser la durée totale d'un tour. Pour un problème de n noeuds à visiter, la relaxation linéaire de cette formulation ne contient que $O(n)$ contraintes et $O(n^2)$ variables. Quelques aspects de l'implantation sont discutés et des expérimentations numériques ont été réalisées pour des problèmes ayant jusqu'à 40 noeuds.

Mots clés: Voyageur de commerce, Fenêtres-horaire.

Introduction

Consider a directed graph $G = (N, A)$, where N is a set of n nodes and A is a set of feasible arcs. Let c_{ij} be the cost of arc (i, j). The traveling salesman problem (TSP) is the problem of finding in a graph G with a given matrix $[c_{ij}]$ a circuit passing through every node once and only once at minimal cost.

Finke *et al.* [1983] have introduced a two-commodity network flow formulation for the TSP. In their formulation, one unit of the first commodity, flow $Y = (y_{ij})$, is delivered at each node while one unit of the second, flow $Z = (z_{ij})$, is picked-up at each node. Hence the total combined amount of the two flows at each node is always the same. Let node 1 correspond to the depot. The model is written as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}(y_{ij} + z_{ij})/(n-1) \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} y_{ij} - \sum_{j=1}^{n} y_{ji} = \begin{cases} n-1 & i = 1 \\ -1 & i \in N \setminus \{1\} \end{cases} \tag{2}$$

$$\sum_{j=1}^{n} z_{ij} - \sum_{j=1}^{n} z_{ji} = \begin{cases} -(n-1) & i = 1 \\ 1 & i \in N \setminus \{1\} \end{cases} \tag{3}$$

$$\sum_{j=1}^{n} (y_{ij} + z_{ij}) = n - 1 \qquad i \in N \tag{4}$$

$$y_{ij} + z_{ij} \in \{0, n-1\} \qquad (i, j) \in A \tag{5}$$

$$y_{ij} \geq 0, \quad z_{ij} \geq 0 \qquad (i, j) \in A. \tag{6}$$

Constraints (2) and (3) define the flow conservation equations for each commodity. Constraints (4) and (5) ensure that there is exactly one arc used out of each node. One can show that model (1–6) can be rewritten equivalently, replacing the flow Y conservation equations (2) by the set of equations

$$\sum_{i=1}^{n} (y_{ij} + z_{ij}) = n - 1 \qquad j \in N. \tag{4'}$$

If one defines $x_{ij} = (y_{ij} + z_{ij})/(n-1)$, then constraints (4) and (4') correspond to the classical assignment constraints. Hence this formulation combines the assignment constraints (4) and (4') with the flow Z conservation equations (3). The reader can also observe that flow conservation equations at the depot node 1 are obtained from the

other equations in constraint sets (2) and (3). Lucena [1986] has considered a similar formulation with various demands d_i, $i = 2, \ldots, n$ at each node. The right-hand sides of equations (2) and (3) are replaced respectively by $-d_i$ and d_i, $i = 2, \ldots, n$, while at the depot node, $d_1 = -\sum_{i=2}^{n} d_i$.

We propose an analogous formulation in which a resource is consumed on each used arc. Let t_{ij} be the amount of resource needed to travel on arc (i, j). Flow Y distributes the resource and flow Z picks it up in such a way that the total combined amount of flows is always the same at each node and equals the total amount T of available resource at the beginning. Essentially, the new formulation defines a variable demand at each node which is given by :

$$d_i = \sum_{j=1}^{n} t_{ji}(y_{ji} + z_{ji})/T \qquad i \in N \setminus \{1\}.$$

When $(y_{ij} + z_{ij})/T = 0$ or 1 for every arc $(i, j) \in A$, the variable demand at node i depends on the sole incoming arc used to reach it.

The quality of the bounds obtained by the linear relaxations of several TSP formulations has been discussed in Langevin *et al.* [1990]. The bounds obtained from the two-commodity flow formulations of Finke *et al.* [1984] and Lucena [1986] are better than the bound of the assigment problem but worst than the bounds obtained from multi-commodity flow formulations (Wong [1980], Claus [1984]) or the equivalent bound provided by the subtour elimination formulation of Dantzig, Fulkerson and Johnson [1954]. The quality of the bound obtained from the linear relaxation of our formulation should compare with the other two-commodity flow formulations although there is no dominance between them.

The formulation that we introduce enables us to handle traveling salesman problems with time windows of up to 40 nodes. In an urban vehicle routing context, this may correspond to an 8-hour work shift for 1 driver. However this formulation is not suited to handle efficiently the classical TSP. We present results on the TSP with time windows and on the makespan problem with time windows.

The paper is organized as follows. Section 1 presents the two-commodity flow formulation for the traveling salesman problem. This formulation is easily extended to the makespan problem. Section 2 shows that this formulation is well suited for problems with time windows; the resource then corresponds to the total amount of available time and the t_{ij}'s correspond to the travel times between nodes. Section 3 discusses implementation issues, and computational results are given in Section 4.

1. A Formulation for the TSP and the Makespan Problem

Let the fixed parameter T be the total amount of resource available at the beginning of any tour. Let node 1 be the starting depot node and for an n-node problem, temporarily define node $n+1$ to be the arrival depot node. This duplication of the depot node is useful to better understand the following formulation with two commodities.

Let t_{ij}, $i, j = 1, 2, \ldots, n+1$ be the amount of resource needed to travel on arc (i, j) and let also c_{ij} be the corresponding cost. Recall that non-negative flow $Y = (y_{ij})$ represents the remaining quantity of resource while non-negative flow $Z = (z_{ij})$ is the accumulated value of resource used. For all arc (i, j), let also $(y_{ij} + z_{ij}) \in \{O, T\}$, that is $(y_{ij} + z_{ij})/T$ be binary variables.

We first present three sets of constraints, their meaning and their reduction. The compact model is presented last.

Constraints on flow Y at each node:

$$\sum_{j=2}^{n} y_{1j} = T$$

$$\sum_{j=2}^{n+1} y_{ij} = \sum_{j=1}^{n} y_{ji} - \sum_{j=1}^{n} t_{ji}(y_{ji} + z_{ji})/T \qquad i = 2, \ldots, n$$

$$\sum_{j=2}^{n} y_{j,n+1} - \sum_{j=2}^{n} t_{j,n+1}(y_{j,n+1} + z_{j,n+1})/T \geq 0.$$

The left-hand side of each relation gives the remaining quantity of resource at each node, for $i = 1, 2, \ldots, n, n+1$. At the start (depot node), the total amount of resource available has value T. For node i, $i = 2, \ldots, n$, the resource available has a value given by the sum of flows on arcs entering node i minus the amount of resource needed to travel on the same arcs. The last relation represents the unused (non-negative) amount of resource at the arrival depot node.

Constraints on flow Z at each node:

$$\sum_{j=2}^{n} z_{1j} = 0$$

$$\sum_{j=2}^{n+1} z_{ij} = \sum_{j=1}^{n} z_{ji} + \sum_{j=1}^{n} t_{ji}(y_{ji} + z_{ji})/T \qquad i = 2, \ldots, n$$

$$\sum_{j=2}^{n} z_{j,n+1} + \sum_{j=2}^{n} t_{j,n+1}(y_{j,n+1} + z_{j,n+1})/T \leq T.$$

The left-hand side of each relation gives the accumulated amount of resource at each node, for $i = 1, 2, \ldots, n, n+1$. At the depot node, flow Z starts with a value of zero. At node i, $i = 2, \ldots, n$, the total amount of flow on arcs entering node i is increased by the amount of resource needed to travel on those arcs. Finally, the total quantity of resource accumulated at the arrival depot node $n+1$ must not exceed the amount T available at the start.

Complementary flow constraints:

$$\sum_{j=2}^{n} (y_{1j} + z_{1j}) = T$$

$$\sum_{j=2}^{n+1} (y_{ij} + z_{ij}) = T \qquad i = 2, \ldots, n$$

$$\sum_{j=2}^{n} (y_{j,n+1} + z_{j,n+1}) = T.$$

At each node, the sum of Y and Z flows is always equal to the total amount of resource at the start.

The next step in the presentation is to replace the first set of constraints on flow Y at each node by an equivalent one. Given that $z_{1j} = 0$, $j = 2, \ldots, n$, the equation at node 1 can be removed as it already appears in the complementary flow constraints. The relation at node $n+1$ can also be discarded since it can be obtained using the two other relations at the same node. Finally, for node i, $i = 2, \ldots, n$, adding both sides of equations on flows Y and Z, and using the corresponding complementary flow constraints leads to

$$\sum_{i=1}^{n} (y_{ij} + z_{ij}) = T \qquad j = 2, \ldots, n.$$

Finally, arcs $(i, n+1)$, $i = 2, \ldots, n$ entering the arrival depot node are renamed to be arcs $(i, 1)$, $i = 2, \ldots, n$. Also, corresponding flow variables Y and Z, resource and cost values are redefined. The formulation for the traveling salesman problem with resource constraints is now given by

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}(y_{ij} + z_{ij})/T \tag{7}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} z_{ij} - \sum_{j=1}^{n} z_{ji} - \sum_{j=1}^{n} t_{ji}(y_{ji} + z_{ji})/T = 0 \qquad i \in N \setminus \{1\} \qquad (8)$$

$$\sum_{i=2}^{n} z_{i1} + \sum_{i=2}^{n} t_{i1}(y_{i1} + z_{i1})/T \leq T \qquad (9)$$

$$\sum_{j=1}^{n} (y_{ij} + z_{ij}) = T \qquad i \in N \qquad (10)$$

$$\sum_{i=1}^{n} (y_{ij} + z_{ij}) = T \qquad j \in N \qquad (11)$$

$$y_{ij} + z_{ij} \in \{0, T\} \qquad (i,j) \in A \qquad (12)$$

$$y_{ij} \geq 0,\ z_{ij} \geq 0 \qquad (i,j) \in A. \qquad (13)$$

The preceding formulation (7–13) corresponds to the classical traveling salesman problem only if T, the fixed amount of available resource, is large enough to travel on the optimal tour. To ensure this, one could choose $T \geq \sum_i \max_j \{t_{ij}\}$. If T is strictly smaller than the total amount of resource needed to travel on the optimal TSP tour, then formulation (7–13) defines a different problem, that is the problem of finding the minimal cost tour using at most T units of the resource.

The linear relaxation of this formulation is obtained by replacing constraints (12) by the inequalities $y_{ij} + z_{ij} \leq T$, $(i,j) \in A$. However, these latter constraints are redundant because of (10) or (11) and can be eliminated. The linear relaxation thus contains $3n$ constraints and $O(n^2)$ variables. Defining variables $x_{ij} = (y_{ij} + z_{ij})/T$, $(i,j) \in A$, the reader can observe that the lower bound provided by the linear relaxation is at least as good as the assignment problem bound. One can also demonstrate that this bound improves for smaller value of T. Another property is that for each node, flow Y defines a path starting at node 1, while flow Z defines a path ending at node 1. Thus the flow supporting graph is connected. However, the solution may contain fractional subtours.

Previously mentioned two-commodity flow formulations are special cases of the proposed one. Finke's formulation is obtained with $t_{i1} = 0$, $i \in N \setminus \{1\}$, $t_{ij} = 1$, $i \in N, j \in N \setminus \{1\}$, and $T = (n-1)$ while Lucena's version is obtained with $t_{i1} = 0$, $i \in N \setminus \{1\}$, $t_{ij} = d_j$, $i \in N$, $j \in N \setminus \{1\}$ and $T = \sum_{j=2}^{n} d_j$. In both cases, constraint (9) becomes redundant.

Finally, in the preceding formulation, if t_{ij} represents the time needed to travel on arc (i,j) and $c_{ij} = t_{ij}$ for all arcs, then the formulation is valid for the Makespan problem (MP) which minimizes the total completion time. In this case, the objective function can equivalently be replaced by the left-hand side of relation (9). Furthermore, the smallest value of the fixed parameter T for which there exists a feasible solution to the linear relaxation is itself a lower bound on the optimal solution value.

2. Formulations for Time Window Constrained Problems

As mentioned above, the TSP formulation with resource constraints is well suited for time window constrained extensions. The resource is the time and constraints specifying the time window restriction at each node are easily formulated on flow Z.

Without loss of generality, let the interval $[0,T]$ be the time window at the depot node, and let $[a_i, b_i]$, $i = 2, \ldots, n$ be the interval defining the earliest time and latest time at which node i can be visited. When needed, let $a_1 = 0$ and $b_1 = T$ at the depot node. This paper deals with the hard time window case, where if the salesman arrives at a node too early, he must wait. In addition, due dates cannot be violated.

Let us define w_i, $i = 2, \ldots, n$ as the waiting time before the visit of node i and c_i, $i = 2, \ldots, n$ as the cost incured per unit of waiting time. Using the same variable and parameter definitions introduced earlier, the formulation of the Traveling salesman problem with time windows (TSPTW) is given by :

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(y_{ij} + z_{ij})/T + \sum_{i=2}^{n} c_i w_i \tag{14}$$

$$\text{s.t. } (9),(10),(11),(12),(13)$$

$$\sum_{j=1}^{n} z_{ij} - \sum_{j=1}^{n} z_{ji} - \sum_{j=1}^{n} t_{ji}(y_{ji} + z_{ji})/T - w_i = 0 \qquad i \in N \setminus \{1\} \tag{15}$$

$$a_i \leq \sum_{j=1}^{n} z_{ij} \leq b_i \qquad i \in N \setminus \{1\} \tag{16}$$

$$w_i \geq 0 \qquad i \in N \setminus \{1\}. \tag{17}$$

Except for the definition of the time window constraints (16) and the introduction of the waiting time variables (17) which appear in (14–15), the constraints have the same meaning as in the TSP model (7–13). The objective function (14) can now take into account the waiting cost incured before servicing the nodes. If the waiting

costs are irrelevant ($c_i = 0$, $i = 2, \ldots, n$), then variables w_i, $i = 2, \ldots, n$ may be simply considered as surplus variables and constraints (15) may therefore be rewritten as inequalities.

If on each arc the cost value is identical to the time needed to travel on the arc, i.e., $c_{ij} = t_{ij}$, $(i,j) \in A$, and if $c_i = 1$, $i = 2, \ldots, n$, then the objective function minimizes the total time on the optimal tour while respecting time windows. This problem is called the Makespan problem with time windows (MPTW). In that case, the objective function can be replaced by the left hand side of constraint (9), that is the arrival time at the depot node.

As in the resource constrained formulation of the TSP, the formulation (9–17) for the TSPTW is valid only if the fixed parameter T is sufficiently large not to eliminate feasible tours. If there exists a solution, the total resource value may be taken as $T \geq \max_{(i,1)\in A}\{b_i + t_{i1}\}$. Relaxing the integrality conditions (12), the TSPTW's linear relaxation formulation involves $O(n^2)$ variables and $4n - 1$ constraints, that is $(n-1)$ new constraints describing the time windows. For the TSPTW, it can be demonstrated that the lower bound also improves with a smaller value of T. For the MPTW, the smallest value of T, feasible for the linear relaxation, is again a lower bound on the optimal total time value.

3. Implementation Issues

Formulations (7–13) and (9–17) all have one redundant constraint amongst the assignment constraints (10–11). Attempting to solve the original LP formulation, without removing one of them, one encounters a lot of numerical instability because the redundant constraint is an equality constraint and any numerical error renders the solution infeasible. In practice one of them is thus removed.

In this section, we introduce a few results on problem preprocessing and describe how the primal lower bound given by the LP relaxation solution is improved by adding violated subtour elimination constraints. In the last part of the section, the branch and bound scheme is described.

3.1. Arc Elimination

In any time window constrained problem, it is very easy to eliminate a portion of the arcs by using the following well known condition: an arc (i,j) exists only if $a_i + t_{ij} \leq b_j$. However, for the single vehicle time window constrained problem, we can derive a much stricter condition. An arc (i,j) can be used in the solution only if all other nodes can

be visited before i or after j. In a more formal way, let us define S_{kl} to be the shortest path from node k to node l. To determine if an arc (i, j) exists, one must check, for all node $l \in N$, if it is possible to either visit node l before node i, i.e., if $a_l + S_{li} \leq b_i$, or to visit node l after node j, i.e., if $a_j + S_{jl} \leq b_l$. If both conditions fail for one node, then arc (i, j) cannot exist. If the depot is one of the nodes i or j, then we need to check only one of the two conditions as it is impossible to visit a node before (after) the starting (arrival) node. We denote by A' the reduced arc set.

3.2. Time Window Reduction

To tighten the LP relaxation and improve the efficiency of the branch and bound algorithm, we reduce the time windows' width using four conditions. These conditions are all based on the following principle. Given a_i the earliest allowed arrival time at node i, let α_i such that $\alpha_i > a_i$ be the earliest possible arrival time of all feasible paths at node i. We can thus tighten the lower bound of the time window at node i to α_i. Similar results hold for the upper bound. The following conditions are applied until no further modifications can be obtained (Desrochers *et al.* 1990).

1) Minimal arrival time from predecessors:

$$a_k = \max\{a_k, \min\{b_k, \min_{(i,k)\in A}\{a_i + t_{ik}\}\}\}. \tag{18}$$

2) Minimal arrival time to successors:

$$a_k = \max\{a_k, \min\{b_k, \min_{(k,j)\in A}\{a_j - t_{kj}\}\}\}. \tag{19}$$

3) Maximal departure time from predecessors:

$$b_k = \min\{b_k, \max\{a_k, \max_{(i,k)\in A}\{b_i + t_{ik}\}\}\}. \tag{20}$$

4) Maximal departure time to successors:

$$b_k = \min\{b_k, \max\{a_k, \max_{(k,j)\in A}\{b_j - t_{kj}\}\}\}. \tag{21}$$

3.3. Subtour Elimination Constraints Detection

The classical subtour elimination constraints for the Traveling salesman problem [Dantzig *et al.* 1954] can be adapted to the various two-commodity flow formulations. We only need to express the subtour elimination constraints using the same variables as our formulation. The original subtour elimination constraints are:

$$\sum_{i,j \in W} x_{ij} \leq |W| - 1 \qquad \forall\, W \subset N, |W| \geq 2. \tag{22}$$

The variables x_{ij} can be expressed as $(y_{ij} + z_{ij})/T$ in our formulations and the subtour elimination constraints become:

$$\sum_{i,j \in W} (y_{ij} + z_{ij}) \leq T(|W| - 1) \qquad \forall\, W \subset N, |W| \geq 2. \tag{23}$$

It should be noted that even though the subtour elimination constraints are verified for any feasible integer solution for the TSP and its extensions, the subtour elimination constraints are not necessarily verified by the LP relaxation of one of these formulations. It is then possible to improve the lower bound obtained using the LP relaxation of the two-commodity flow formulation by detecting violated subtour elimination constraints and by adding them to the LP relaxation.

The identification of violated subtour elimination constraints is a well solved problem (see Padberg and Groetschel 1985). In a first phase, we project the LP relaxation solution onto the x_{ij} variables using $x_{ij} = (y_{ij} + z_{ij})/T$. A shrinking procedure [Padberg and Groetschel 1985] is then applied to reduce the size of the graph. The shrinking procedure uses the fact that if $x_{ij} + x_{ji} = 1$, and if there is a violated subtour such that $i \in W$, then there also exists another violated subtour such that $i, j \in W$. Quite often, some violated subtour inequalities are detected during the shrinking algorithm. In a second phase, we use the Gomory and Hu [1961] algorithm to find the minimum cut tree on the reduced graph. The detection of violated subtour inequalities is done in polynomial time.

3.4. The Branch and Bound Scheme

The lower bounds in our branch and bound scheme are obtained by solving the LP relaxation of the two-commodity flow formulation using the XMP software (Marsten 1981) and by adding any violated subtour elimination constraints. Of course, for the time constrained versions, the first step at any node of the branch and bound tree is to preprocess the problem using arc elimination rules and time window reduction. At

a given node, an arc $(i,j) \in A$ is chosen for branching. Arc (i,j) is either fixed to zero or forced to be included in the solution. In the time constrained problems, this is done by setting $(y_{ij} + z_{ij}) = 0$ or $(y_{ij} + z_{ij}) = T$. In the other cases, this is done by setting $(y_{ij} + z_{ij} + y_{ji} + z_{ji}) = 0$ or otherwise $(y_{ij} + z_{ij} + y_{ji} + z_{ji}) = T$. The arc chosen for branching is the one with a fractional value closest to 1/2. At any stage, the branch and bound tree node chosen for exploration is the unexplored node having the best lower bound. This branch and bound strategy is similar to that of Padberg and Rinaldi [1987] for the Symmetric TSP.

Throughout the search, at a given branch and bound tree node, subtour elimination constraints previously introduced are not to be removed from the current LP program unless they become redundant. This is clearly valid since those constraints define valid inequalities for the the two-commodity formulation.

4. Computational experiments

4.1. Description of the Test Problems

In order to have some empirical knowledge on the validity of our algorithm, we tested our branch and bound algorithm on some randomly generated problems. The problems were generated using three distance or cost classes: Asymmetric Random (AR), Symmetric Random (SR) and Symmetric Euclidean (SE). The distance matrices were rounded to the nearest tenth and were generated using the following schemes:

AR: $c_{ij} \sim U[0,100], i \neq j$;

SR: $c_{ij} \sim U[0,100], i < j$;

SE: $c_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{\frac{1}{2}}$, with $(x_i, y_i) \sim U[0,50]^2, i < j$.

The travel time matrix is taken to be identical to the cost matrix. A second nearest neighbor tour is found. This tour is used as an upper bound for the TSP and as starting point to generate the time windows for the TSPTW and the MPTW. Let the tour $(\pi_1, \pi_2, \ldots, \pi_n, \pi_1)$ with $\pi_1 = 1$ be the second nearest neighbor tour. The visiting time at each node is determined in the following way:

$$v_{\pi_1} = 0$$
$$v_{\pi_k} = v_{\pi_{k-1}} + t_{\pi_{k-1},\pi_k} + U[0, wait], \qquad k = 2, \ldots, n-1.$$

Parameter *wait* has been set to the value 10 in ten percent of the makespan with time windows problems, and to zero in the other cases. Also notice that there are no waiting

costs for the TSPTW tests, that is $c_i = 0$ for all $i \in N$, while $c_i = 1$ for all $i \in N$ in the MPTW tests. The time windows are then generated around the visiting time at each node as follows:

$$\begin{aligned} a_{\pi_1} &= 0 \\ a_{\pi_k} &= v_{\pi_k} - U[0, width], \qquad k = 2, \ldots, n-1, \\ b_{\pi_k} &= v_{\pi_k} + U[0, width], \qquad k = 2, \ldots, n-1, \\ b_{\pi_1} &= b_{\pi_n} + t_{\pi_n, \pi_1}. \end{aligned}$$

For each problem type (TSPTW and MPTW), for each value of the parameter *width* (20, 30 and 40) and for each distance class (AS, SR and SE), ten 20, 30 and 40-node problems where generated. Main results appear in tables 1 and 2. All the numbers represent the average values calculated over the number of successes (S), that is the number of problems which have been completely solved.

4.2. Numerical Results

It should be first noted that the average time window widths are not presented since they essentially are equal to the parameter value. The time window reduction has a negligible effect at the begining; however it but becomes important in the branch and bound exploration tree.

The second remark is that, comparing $|A|$ and $|A'|$, the size of the arc set is dramatically reduced by the arc elimination procedure, in both problem types. Given the time window sizes considered, the number of arcs is mostly a linear function of the number of nodes.

Next, four primal bounds are presented comprising the upper bound value T and the values of three lower bounds: (ASP) the value of the assignment problem solution on the same graph, (TCF) the value of the LP relaxation of the two-commodity formulation, and the value of the LP relaxation of the two-commodity flow formulation plus any violated subtour elimination constraints (SEC). The T value does not have a strong effect on the solution process, except that it is an essential parameter of the formulation. It is always far beyond the optimal value OPT. The reader can observe that the assignment bound (ASP) is very close to the two-commodity flow formulation (TCF) in the TSPTW cases, but up to 10% gap on the MPTW. The upgrade on the TCF bound given by the subtour elimination constraints (SEC) is clearly significant in the TSPTW cases while it is not for the MPTW. Evidently, if the gap between SEC and OPT is large, the number of nodes in the branch and bound tree is important, as well as the CPU time given in seconds on a SUN SPARK work station.

In general, random asymmetric test problems where easily solved for both types of problems. Consequently, it is possible to augment the problem size and the proposed formulation and algorithm are still adequate. In the random symmetric cases, only 5 out of 180 test problems where not solved to optimality. Fine exploration of the results by the authors indicates that either the number of branch and bound nodes exceeds 200 (3 cases) or numerical instability appears at some node of the exploration tree. Random symmetric problems are however relatively easy to solve, but they take much more CPU time. Finally, symmetric euclidean problems are the most difficult to solve, specially for the makespan type. Careful re-examination of the results indicates that the tree search size has been exceeded or that numerical instability appears at some stage.

Conclusion

We have introduced a new formulation for the Traveling salesman problem based on a two-commodity flow network. This resource constrained formulation is well-suited to handle time window constraints and easily extended to makespan type problems.

Numerical experimentation shows that this formulation allows solving time constrained traveling salesman problems of up to 40 nodes. This size of problems covers a large part of the real life problems. In practice, a vehicle rarely visits more than 40 customers in an 8-hour work shift. This would correspond to a customer every 10 minutes, which is a very high level of productivity considering the time needed for actual service at each customer's location and the low speed of the vehicles in large cities.

This resource constrained formulation is compact and it allows us to handle problems with other special constraints. It can be easily adapted to consider precedence relations and delays between service at different nodes. Various versions of the formulation can take into account other types of constraints: vehicle capacity constraints with pick-ups at some nodes and deliveries at others, resource limitations on arcs or nodes, etc. Further researches are however needed to solve more efficiently this new type of formulation, using for example, decomposition approaches or additive bounding procedures.

Acknowledgments

We would like to thank Mr Olivier Bahn and Ms Sylvie Gélinas who performed the computational study. The authors were supported by grants from the Canadian Government (NSERC) and the Québec Government (FCAR).

References

Claus, A., (1954). A New Formulation for the Travelling Salesman Problem, *SIAM J. Alg. Disc. Math.* **5**, 21–25.

Dantzig, G., Fulkerson, D. and Johnson, S., (1954) Solutions of Large-Scale Traveling-Salesman Problem. *Operations Research* **2**, 393-410.

Desrochers, M., Desrosiers, J. and Solomon, M., (1990). *A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows.* Cahiers du GERAD G–90–30, 13pp. To appear in *Operations Research.*

Finke, G., Claus, A. and Gunn, E., (1984). A Two-Commodity Network Flow Approach to the Traveling Salesman Problem. *Congressus Numerantium 41,* 167-178.

Gomory, R.E. and Hu, T.C., (1961). Multi-Terminal Network Flows, *SIAM Journal 9,* 551–570.

Langevin, A., Soumis, F. and Desrosiers J., (1990). Classification of Traveling Salesman Problem Formulations, *Operations Research Letters 2,* 393–410.

Lucena, A., (1986). *Exact Solution Approaches for the Vehicle Routing Problem.* Ph. D. thesis, Management Science Dept., Imperial College, London.

Marsten, R.E., (1981). The Design of the XMP Linear Programming Library. *ACM Transact. Math. Software 7,* 481-497.

Padberg, M. and Groetschel, M., (1985). Polyhedral Computations in E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Schmoys, eds. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* Wiley.

Padberg, M. and Rinaldi, G., (1987). Optimization of a 532-City Symmetric Traveling Salesman Problem by Branch and Cut. *Operations Research Letters 6,* 1-7.

Wong, R.T., (1980). Integer Programming Formulations of the Travelling Salesman Problem, *Proceedings of the IEEE International Conference of Circuits and Computers*, 149–152.

ASYMMETRIC RANDOM (AR)

n	WIDTH	S	$\|A\|$	$\|A'\|$	T	ASP	TCF	SEC	OPT	BB	CPU
20	20	10	192.0	22.3	553.2	542.8	542.8	542.8	542.8	1.0	1.3
	30	10	198.7	26.9	558.3	540.0	540.0	540.0	540.4	1.6	1.8
	40	10	205.0	32.2	563.5	535.2	535.2	535.7	536.4	1.6	2.5
30	20	10	428.6	35.3	621.6	608.6	608.6	608.6	608.6	1.0	2.4
	30	10	441.4	43.7	627.3	598.9	598.9	598.9	598.9	1.0	3.3
	40	10	455.9	59.7	633.0	574.8	574.8	577.4	577.7	1.2	4.9
40	20	10	738.7	52.9	664.2	641.6	641.6	642.3	642.3	1.0	4.5
	30	10	768.9	77.5	670.1	640.1	640.1	640.8	640.8	1.0	7.3
	40	10	797.4	129.9	676.9	616.2	616.3	618.9	626.7	2.2	16.7

SYMMETRIC RANDOM (SR)

n	WIDTH	S	$\|A\|$	$\|A'\|$	T	ASP	TCF	SEC	OPT	BB	CPU
20	20	10	195.6	27.0	545.1	526.3	526.3	528.3	528.3	1.0	1.9
	30	10	204.0	36.0	551.1	511.8	511.9	517.7	526.4	1.4	2.8
	40	10	210.2	44.8	557.3	493.5	493.6	501.9	502.0	1.2	3.6
30	20	10	432.3	49.5	628.0	595.5	595.6	597.5	605.4	2.6	6.0
	30	10	446.4	66.0	632.9	587.5	587.6	592.6	597.1	1.2	5.8
	40	10	460.2	91.9	637.4	580.1	580.4	591.4	592.3	2.4	10.4
40	20	10	753.9	82.8	651.6	602.0	602.6	606.6	616.4	7.0	35.7
	30	9	783.2	111.0	658.3	595.7	595.8	603.2	610.2	18.6	139.8
	40	10	813.0	173.6	663.0	579.3	589.5	598.9	606.8	31.0	398.2

SYMMETRIC EUCLIDEAN (SE)

n	WIDTH	S	$\|A\|$	$\|A'\|$	T	ASP	TCF	SEC	OPT	BB	CPU
20	20	10	217.0	39.0	342.3	318.4	318.5	320.3	321.3	2.0	3.1
	30	10	223.9	58.9	348.1	308.2	308.4	313.7	317.6	8.8	12.3
	40	10	232.4	80.8	353.6	293.1	293.6	306.3	307.9	8.2	20.6
30	20	9	481.1	75.3	448.4	414.2	414.3	417.6	419.0	4.3	11.4
	30	9	497.9	119.4	449.6	389.6	389.9	401.3	405.4	13.4	47.4
	40	4	513.5	161.3	452.8	358.1	358.5	371.2	375.4	31.5	133.9
40	20	9	851.0	126.9	540.9	495.7	495.7	500.1	503.1	10.3	43.6
	30	8	877.8	199.4	536.1	454.9	455.1	465.5	468.0	8.5	53.7
	40	9	903.4	267.4	545.9	450.2	450.5	465.2	469.7	61.7	710.4

Table 1: The Traveling Salesman Problem with Time Windows (TSPTW)

ASYMMETRIC RANDOM (AR)

n	WIDTH	S	$\|A\|$	$\|A'\|$	T	ASP	TCF	SEC	OPT	BB	CPU
20	20	10	192.6	22.2	559.3	542.8	545.3	545.3	545.3	1.4	1.8
	30	9	198.9	26.0	559.1	537.9	539.2	539.2	539.6	1.7	2.2
	40	10	205.6	32.8	569.6	534.7	543.6	543.6	543.7	2.8	3.4
30	20	10	429.4	35.9	637.1	608.6	619.8	619.8	619.8	2.0	3.7
	30	10	442.5	44.5	642.7	598.9	609.7	609.7	614.2	2.6	5.5
	40	10	456.6	61.4	648.6	574.8	594.6	594.6	611.1	7.6	16.6
40	20	10	744.9	52.8	683.5	645.1	661.7	661.7	661.7	2.0	5.8
	30	10	773.4	79.3	689.4	640.1	660.9	660.9	660.9	8.0	27.6
	40	10	798.8	124.9	696.0	616.2	648.4	648.4	654.7	14.4	80.2

SYMMETRIC RANDOM (SR)

n	WIDTH	S	$\|A\|$	$\|A'\|$	T	ASP	TCF	SEC	OPT	BB	CPU
20	20	10	197.4	27.2	557.3	526.3	540.9	540.9	540.9	2.2	2.2
	30	10	204.1	34.7	563.1	512.5	539.4	539.4	539.4	3.4	3.7
	40	10	211.3	46.2	569.9	493.5	535.1	537.0	537.5	9.4	14.4
30	20	10	434.4	50.2	640.9	595.5	619.4	619.4	624.9	6.4	9.9
	30	10	447.8	66.5	645.7	587.1	618.4	618.4	623.8	14.6	30.5
	40	10	461.6	91.7	650.4	580.1	617.1	617.4	623.2	38.4	152.2
40	20	10	759.6	83.8	669.9	602.0	627.9	627.9	633.1	14.2	45.2
	30	9	787.1	115.2	675.9	594.7	625.7	625.7	634.3	52.8	338.9
	40	7	820.9	159.0	692.4	593.9	631.9	631.9	641.6	84.1	744.8

SYMMETRIC EUCLIDEAN (SE)

n	WIDTH	S	$\|A\|$	$\|A'\|$	T	ASP	TCF	SEC	OPT	BB	CPU
20	20	10	216.4	37.2	352.0	318.8	331.4	331.9	332.8	9.8	7.7
	30	10	223.0	56.4	357.5	308.6	328.8	328.8	330.5	30.4	35.8
	40	5	228.2	68.2	363.8	300.4	328.7	329.3	329.3	36.2	56.4
30	20	9	481.0	75.6	460.6	409.3	443.9	443.9	443.9	50.1	87.9
	30	6	494.2	109.5	481.0	404.7	454.4	454.4	454.4	107.0	353.1
	40	2	507.5	142.0	500.5	385.3	465.4	465.4	465.4	137.0	567.5
40	20	7	850.1	126.6	544.3	476.3	528.4	528.4	528.4	87.9	299.6
	30	0	–	–	–	–	–	–	–	–	–
	40	0	–	–	–	–	–	–	–	–	–

Table 2: The Makespan Problem with Time Windows (MPTW)